

## REMARKS

### I. Introduction

Claims 1 to 19, and 40 to 62 are currently pending in the present application. Applicants appreciate the Examiner's indication that claims 8 and 47 would be allowable if placed in independent form.

In view of the following remarks, it is respectfully submitted that the remainder of the presently pending claims are allowable, and reconsideration of the present application is respectfully requested.

### II. Rejection of Claims 1 to 7, 9 to 19, 40 to 46 and 48 to 62 Under 35 U.S.C. § 103(a)

Claims 1 to 7, 9 to 19, 40 to 46 and 48 to 62 stand rejected under 35 U.S.C. § 103(a) as unpatentable over the combination of U.S. Patent No. 5,844,918 ("Kato"), U.S. Patent No. 7,088,398 ("Wolf") and that which the Examiner refers to as "Applicants Admitted Prior Art (AAPA)." Without addressing or agreeing with the Examiner's characterization of any of Applicants' disclosure as constituting an admission of prior art, it is respectfully submitted that the combination of Kato, Wolf and AAPA does not render unpatentable any of the present claims for at least the following reasons.

Claim 1 relates to a method for detecting errors in a data package, and claim 40 relates to an article of manufacture comprising a computer-readable medium having stored thereon instructions adapted to be executed by a processor, the instructions which, when executed, define a series of steps to be used to detect errors in a data package. Each of claims 1 and 40 has been amended to recite the following:

*... receiving at least two data elements; receiving, separately and at a different time from the at least two data elements, a set of desired code point values corresponding to a data package; determining a set of current code point values for the at least two data elements; and comparing the set of current code point values to the set of desired code point values.*

Kato provides for transmission of data packets having appended thereto error detecting codes, *i.e.*, the error detecting codes (relied upon by the Examiner as allegedly disclosing the recited code point values) are sent with the data packets (relied upon by the Examiner as allegedly disclosing the recited at least two data elements). The Office Action

admits that the combination of Kato does not disclose the feature of “receiving, separately and at a different time from the at least two data elements, a set of desired code point values corresponding to a data package.”

However, the Examiner asserts that it would have been obvious to combine Kato with Wolf and that Wolf describes transmitting the data and code point values/error detection code separately at different times. Leaving aside the issue of whether the combination of Kato and Wolf is appropriate, Applicants respectfully disagree with the Examiner’s assertion that Wolf discloses receiving data and code point values separately and at different times. Wolf describes a serial link for sending encoded video and one or more other streams of data (e.g., encoded audio) over one or more channels. Video data is sent during active video periods and other data is sent during “data islands” (a time interval that neither coincides with nor overlaps the active video period). *See, e.g.*, 7:15-34. Wolf describes sending each packet such that the packet is spread over three channels and each subpacket of the packet is spread over two channels. BCH parity bits are also separately generated for each subpacket and transmitted over the two channels. *See, e.g.*, 39:9-17. However, as described at 34:18-44, the BCH parity bits, while *generated* separately from the subpackets, are sent *with* the subpackets, not *separately* from the subpackets: “Each subpacket includes 56 data bits (i.e., 7 bytes, each byte corresponding to one 10-bit TMDs code word) followed by 8 BCH parity bits.” While the subpacket is spread over two channels, the BCH parity bits are still transmitted with the subpacket. *See, e.g.*, figure 9.

Since the BCH parity bits of Wolf are received together with and at the same time as the subpackets that they are transmitted with, Wolf does not disclose “receiving, separately and at a different time from the at least two data elements, a set of desired code point values corresponding to a data package.”

Accordingly, the combination of Kato, Wolf and AAPA does not disclose or suggest the features of either of claims 1 and 40, so the combination of Kato, Wolf and AAPA does not render unpatentable either of claims 1 and 40.

Claim 12 relates to a method for detecting changes in a data package transmitted over a network, and claim 51 relates to an article of manufacture comprising a computer-readable medium having stored thereon instructions adapted to be executed by a processor, the

instructions which, when executed, define a series of steps to be used to detect changes in a data package transmitted over a network. Each of claims 12 and 51 recites the following:

*... receiving, from a first network component, a set of current code point values corresponding to a data package at a code point monitor; receiving, from a second network component, a set of desired code point values corresponding to the data package at the code point monitor; and comparing the set of current code point values to the set of desired code point values, wherein the set of current code point values is determined at the first network component based on the data package received by the first network component.*

The Office Action does not address these features recited in claims 12 and 51.

The Office Action also admits that Kato does not disclose the feature of “receiving, from a second network component, a set of desired code point values corresponding to the data package at the code point monitor.” However, the Examiner asserts that it would have been obvious to combine Kato with Wolf and that Wolf describes “transmitting the data and code point values/ error detection code separately and at different times.” As discussed above, Wolf does not teach receiving data and code point values separately and at different times, but even assuming that the Examiner’s assertion was correct, claims 12 and 51 require receiving a set of current code point values from a first network component and receiving a set of desired code point values from a second network component. Wolf and Kato describe receiving both the data elements and the code point values from the same network component. See, e.g., Kato at Figure 1 and Wolf at Figure 2. Applicant respectfully asserts that the examiner’s rejection does not address these features of claims 12 and 51 and furthermore that Wolf does not describe these features.

Accordingly, the combination of Kato, Wolf and AAPA does not disclose or suggest the features of either of claims 12 and 51, so that the combination of Kato, Wolf and AAPA does not render unpatentable either of claims 12 and 51.

Claim 62 relates to a method for detecting errors in a data package and recites the following:

*... receiving a set of desired code point values corresponding to a data package; storing the set of desired code point values; after the storing of the set of desired code point values, receiving at least two data elements; determining a set of current code point values for the at least two data elements; and comparing the set of current code point values to the stored set of desired code point values.*

The Office Action does not address these features recited in claim 62 but rather largely repeats the text of the rejections given for claims 12 and 51. As discussed above, Wolf does not teach receiving data and code point values separately, but even assuming that Wolf did so teach, claim 62 requires “receiving a set of desired code point values corresponding to a data package; storing the set of desired code point values; after the storing of the set of desired code point values, receiving at least two data elements; determining a set of current code point values for the at least two data elements; and comparing the set of current code point values to the stored set of desired code point values.” Applicant respectfully asserts that the examiner’s rejection does not address these features of claim 62 and furthermore that Wolf does not describe these features because, as discussed above, Wolf describes receiving parity bits with the subpacket data thus the parity bits cannot be stored before the subpacket data is received.

Accordingly, the combination of Kato, Wolf and AAPA does not disclose or suggest the features of claim 62, so that the combination of Kato, Wolf and AAPA does not render unpatentable claim 62.

As for claims 2 to 11 and 59 to 61, which ultimately depend from claim 1 and therefore include all of the features recited in claim 1; claims 13 to 19, which ultimately depend from claim 12 and therefore include all of the features recited in claim 12; claims 41 to 50, which ultimately depend from claim 40 and therefore include all of the features recited in claim 40; and claims 52 to 58, which ultimately depend from claim 51 and therefore include all of the features recited in claim 51, it is respectfully submitted that the combination of Kato, Wolf and AAPA does not render unpatentable these dependent claims for the same reasons set forth above in support of the patentability of their respective base claims. *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596 (Fed. Cir. 1988) (any dependent claim that depends from a non-obvious independent claim is non-obvious).

Withdrawal of this rejection is therefore respectfully requested.

**III. Conclusion**

In light of the foregoing, it is respectfully submitted that all pending claims are in condition for allowance. Prompt reconsideration and allowance of the present application are therefore earnestly solicited.

Please charge any fees due, or credit any overpayments, to Kenyon & Kenyon LLP Deposit Account 11-0600.

Respectfully submitted,

Dated: May 27, 2008

/Paul T. Qualey/

Paul T. Qualey  
Reg. No. 45,027

KENYON & KENYON LLP  
1500 K Street, N.W., Suite 700  
Washington, DC 20005-1257  
(202) 220-4200  
**CUSTOMER NO. 23838**